Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously presented) A computer-implemented character validation method comprising the steps of:

retrieving a data value from a character stream; and

determining a validity of a character represented by said data value by locating a member of a data structure, said member having a direct correspondence to said data value, wherein said validity is determined according to a logical combination of a plurality of status values in said member of said data structure, wherein the determining step determines the data value's validity as a character within a given computer language.

2. (Previously presented) The computer-implemented method of claim 1 wherein said data structure comprises an array and further comprising the step of indexing into said array using said data value, wherein a member of said array corresponding to said data value is pointed to in response to said indexing step.

3. (Canceled)

4. (Previously presented) The computer-implemented method of claim 1 wherein, if the logical combination corresponds to a logically "TRUE" value, said data value represents a valid character.

5. (Previously presented) The computer-implemented method of claim 1 further comprising the step of, if each character in said character stream is valid, applying a predetermined set of syntactic rules to byte patterns comprising said character stream.

6. (Previously presented) The computer-implemented method of claim 1 further comprising the step of generating said data structure.

7. (Previously presented) The computer-implemented method of claim 5 wherein said character stream comprises characters in accordance with a specification for an extensible markup language, and wherein said status values are set in accordance with a set of valid characters defined in said specification.

8. (Previously presented) The computer-implemented method of claim 7 wherein the extensible markup language comprises XML and wherein said syntactic rules include rules in accordance with XML.

9. (Previously presented) A data processing system comprising:

first circuitry operable for retrieving a data value from a character stream; and

second circuitry operable for determining a validity of a character represented by said data value by locating a member of a data structure, said member having a direct correspondence to said data value, wherein said validity is determined according to a logical combination of a plurality of status values in said member of said data structure, wherein the second circuitry determines the data value's validity as a character within a given computer language.

10. (Previously presented) The system of claim 9 wherein said data structure comprises an array and further comprising circuitry operable for indexing into said array using said data value, wherein a member of said array corresponding to said data value is pointed to as part of the operation of said second circuitry.

11. (Canceled)

12. (Original) The system of claim 9 wherein, if said logical combination corresponds to a logically "TRUE" value, said data value represents a valid character.

13. (Previously presented) The system of claim 9 further comprising third circuitry operable for, if each character in said character stream is valid, applying a predetermined set of syntactic rules to byte patterns comprising said character stream.

14. (Previously presented) The system of claim 9 further comprising fourth circuitry operable for generating said data structure.

15. (Original) The system of claim 13 wherein said character stream comprises characters in accordance with a specification for an extensible markup language, and wherein said status values are set in accordance with a set of valid characters defined in said specification.

16. (Original) The system of claim 15 wherein the extensible markup language comprises XML and wherein said syntactic rules include rules in accordance XML.

17. (Previously presented) A computer program product embodied in a machine-readable storage medium including programming for validation, the programming comprising a set of instructions for performing the steps of:

retrieving a data value from a character stream;

determining a validity of a character represented by said data value by locating a member of a data structure, said member having a direct correspondence to said data value, wherein said validity is determined according to a logical combination of a plurality of status values in said member of said data structure, wherein the determining step determines the data value's validity as a character within a given computer language.

18. (Previously presented) The program product of claim 17 wherein said data structure comprises an array and further comprising instructions for performing the step of indexing into said array using said data value, wherein a member of said array corresponding to said data value is pointed to in response to said indexing step.

19. (Canceled)

20. (Original) The program product of claim 17 wherein, if the logical combination corresponds to a logically "TRUE" value, said data value represents a valid character.

21. (Previously presented) The program product of claim 17 further comprising instructions for performing the step of, if each character in said stream is valid, applying a predetermined set of syntactic rules to byte patterns comprising said character stream.

22. (Original) The program product of claim 17 further comprising the step of generating said data structure.

23. (Original) The program product of claim 21 wherein said character stream comprises characters in accordance with a specification for an extensible markup language, and wherein said status values are set in accordance with a set of valid characters defined in said specification.

24. (Original) The program product of claim 23 wherein the extensible markup language comprises XML and wherein said syntactic rules include rules in accordance with XML.

25. (Previously presented) A character validation method comprising the steps of:

retrieving a data value from a character stream;

determining a validity of a character represented by said data value by locating a member of a data structure, said member having a direct correspondence to said data value, wherein said validity is determined according to a logical combination of a plurality of status values in said member of said data structure, wherein said character stream comprises characters in accordance with a specification for an extensible markup language, and wherein a first status value of said plurality of status values indicates whether said data value represents a valid character having a first attribute corresponding to said first status value and a second status value of said plurality of status values indicates whether said data value represents a valid character having a second attribute corresponding to said second status value, wherein the determining step determines the data value's validity as a character within a given computer language; and

if each character in said stream is valid, applying a predetermined set of syntactic rules to byte patterns comprising said character stream in accordance with said extensible markup language.

26. (Previously presented) The method of claim 25 wherein said character stream comprises a message packaged in accordance with a an extensible markup language, said first status value indicates whether said data value is a valid base character, said second status value indicates whether said data value is a valid digit character, and a third status value indicates whether said data value is a valid extender character.

27. (Previously presented) The computer-implemented character validation method of claim 1 wherein said character stream comprises characters in accordance with a specification for an extensible markup language and within said plurality of status values, a first status value indicates whether said data value represents a valid base character, a second status value indicates whether said data value represents a valid digit character, and a third status value indicates whether said data value is a valid extender character.

28. (Previously presented) The data processing system of claim 9 wherein said character stream comprises characters in accordance with a specification for an extensible markup language and within said plurality of status values, a first status value indicates whether said data value represents a valid base

character, a second status value indicates whether said data value represents a valid digit character, and a third status value indicates whether said data value is a valid extender character.

29. (Previously presented) The computer program product of claim 17 wherein said character stream comprises characters in accordance with a specification for an extensible markup language and within said plurality of status values, a first status value indicates whether said data value represents a valid base character, a second status value indicates whether said data value represents a valid digit character, and a third status value indicates whether said data value is a valid extender character.